

In the claims:

1. (currently amended) A method for ~~routing processing a data packet plurality of independent multi-packet threads~~ comprising:

producing a plurality of threads associated with the packet, each thread being a sequence of instructions that facilitates packet routing and that is independently executable with respect to other ones of the threads;

assigning a thread identifier (TID) to each of the ~~independent multi-IP packet threads~~ and maintaining an activity status for each thread;

for each thread, selecting a pipeline from a plurality of pipelines, at least some of which are specialized, and forwarding that thread to the selected pipeline, such that processing of each packet is divided into multiple independent threads which are processed by multiple pipelines, and such that delay in processing of a first packet routing thread in a first pipeline does not affect processing of a second packet routing thread in a second pipeline

~~processing a first multi-IP packet thread in a first stage of a processing pipeline; and
— responsive to an activity status of the first multi-IP packet thread, forwarding the first multi-IP packet thread to a next stage of the processing while forwarding a second multi-IP packet thread to the first stage of the processing pipeline such that the first and the second multi-IP packet threads can be processed simultaneously in the processing pipeline, and wherein the independence of the multi-IP packet threads eliminates IP packet processing delays.~~

2. (currently amended) The method for ~~processing the plurality of independent multi-Internet Protocol (IP) packet threads~~ according to claim 1, further comprising: transferring the first ~~multi-IP packet thread~~ from an input buffer to a packet task manager; dispatching the first ~~multi-IP packet thread~~ from the packet task manager to an analysis machine; classifying the first ~~multi-IP packet thread~~ in the analysis machine; and modifying and forwarding the first ~~multi-IP packet thread~~ in a packet manipulator.

3. (currently amended) The method for ~~processing the plurality of independent multi-IP packet threads~~ according to claim 1, ~~wherein the activity status~~, wherein the activity status indicates that a status of the associated ~~multi-IP packet thread~~ is one of active, inactive or waiting.

4. (cancelled)

5. (currently amended) An apparatus for ~~routing processing~~ a plurality of independent multi-Internet Protocol (IP)-packet threads, said apparatus comprising:

a memory for storing:

a plurality of threads associated with the packet, each thread being a sequence of instructions that facilitates packet routing and that is independently executable with respect to other ones of the threads;

a unique Thread Identifier (TID) for each thread; and

an activity status for each thread; and

an analysis machine including a plurality of pipelines, at least some of which are specialized, the analysis machine selecting a pipeline for each thread and forwarding that thread to the selected pipeline such that processing of each packet is divided into multiple independent threads which are processed by multiple pipelines, and such that delay in processing of a first packet routing thread in a first pipeline does not affect processing of a second packet routing thread in a second pipeline

~~a processing pipeline including a plurality of stages coupled to receive and process the plurality of independent multi-IP packet threads such that, during a processing period, each of the plurality of stages of the processing pipeline is operating on a different one of the multi-IP packet threads from the plurality of multi-IP packet threads, and wherein the independence of the multi-IP packet threads eliminates pipeline processing delays; and~~

~~storage for storing data associated with each of the multi-IP packet threads, the data including a Thread Identifier (TID) and an activity status for each of the multi-IP packet threads, wherein processing pipeline selectively forwards the multi-IP packet threads through the processing pipeline in response to the activity status of the multi-IP packet thread.~~

6. (currently amended) The apparatus according to claim 5, ~~further comprising; an analysis machine having multiple pipelines, wherein one pipeline is dedicated to directly manipulating individual data bits of a bit field, and further comprising~~ ÷ a packet task manager operationally

connected to said analysis machine, \div and $\frac{1}{\div}$ a packet manipulator operationally connected to said analysis machine.

7. (original) The apparatus according to claim 6, wherein said analysis machine is multi-threaded.

8. (original) The apparatus according to claim 6, wherein said analysis machine has 32 threads.

9. (original) The apparatus according to claim 6, further comprising: a packet task manager operationally connected to said analysis machine; a packet manipulator operationally connected to said analysis machine; and a global access bus including a master request bus and a slave request bus separated from each other and pipelined.

10. (original) The apparatus according to claim 6, further comprising: an external memory engine operationally connected to said analysis machine; and a hash engine operationally connected to said analysis machine.

11. (previously presented) The apparatus according to claim 9, further comprising: packet input global access bus program code, stored in a computer readable memory and operable when executed to control a flow of data packet information from a flexible input data buffer to the analysis machine.

12. (previously presented) The apparatus according to claim 9, further comprising: packet data global access bus program code, stored in a computer readable memory and operable when executed to control a flow of packet data between a flexible data input bus and the packet manipulator.

13. (previously presented) The apparatus according to claim 9, further comprising: statistics data global access bus software code used for connection of the analysis machine to the packet manipulator.

14. (previously presented) The apparatus according to claim 9, further comprising: private data global access bus software code used for connection of the analysis machine to an internal memory engine submodule.

15. (previously presented) The apparatus according to claim 9, further comprising: lookup global access bus software code used for connection of the analysis machine to an internal memory engine submodule.

16. (original) The apparatus according to claim 9, further comprising: results global access bus software code used for providing flexible access to an external memory.

17. (previously presented) The apparatus according to claim 5, wherein the activity status indicates that the associated multi-IP packet thread status is one of active, inactive or waiting.

18. (original) The apparatus according to claim 9, further comprising: a bi-directional access port operationally connected to said analysis machine; an input buffer operationally connected to said analysis machine; and an output buffer operationally connected to said analysis machine.